

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re Patent Application of:

Hideki SAKURAI et al.

Application No.:

Group Art Unit:

Filed:

Examiner:

For: APPARATUS AND METHOD FOR APPLYING REVISION INFORMATION TO
SOFTWARE

**SUBMISSION OF CERTIFIED COPY OF PRIOR FOREIGN
APPLICATION IN ACCORDANCE
WITH THE REQUIREMENTS OF 37 C.F.R. § 1.55**

Commissioner for Patents
PO Box 1450
Alexandria, VA 22313-1450

Sir:

In accordance with the provisions of 37 C.F.R. § 1.55, the applicant(s) submit(s) herewith
a certified copy of the following foreign application:

Japanese Patent Application No(s). 2003-329007

Filed: September 19, 2003

It is respectfully requested that the applicant(s) be given the benefit of the foreign filing
date(s) as evidenced by the certified papers attached hereto, in accordance with the
requirements of 35 U.S.C. § 119.

Respectfully submitted,

STAAS & HALSEY LLP

Date: 3/26/04

By: Richard A. Gollhofer
Richard A. Gollhofer
Registration No. 31,106

1201 New York Ave, N.W., Suite 700
Washington, D.C. 20005
Telephone: (202) 434-1500
Facsimile: (202) 434-1501

JAPAN PATENT OFFICE

This is to certify that the annexed is a true copy of the following application as filed with this office.

Date of Application: September 19, 2003

Application Number: Patent Application No. 2003-329007
[ST.10/C] [JP2003-329007]

Applicant(s): FUJITSU LIMITED.

December 22, 2003

Commissioner,

Japan Patent Office Yasuo IMAI

Certificate No.P2003-3106431

日本国特許庁
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出願年月日 2003年 9月19日
Date of Application:

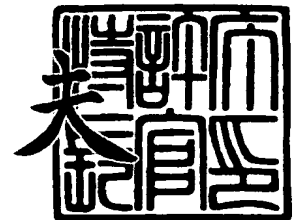
出願番号 特願2003-329007
Application Number:
[ST. 10/C]: [JP 2003-329007]

出願人 富士通株式会社
Applicant(s):

2003年12月22日

特許庁長官
Commissioner,
Japan Patent Office

今井 康



出証番号 出証特2003-3106431

【書類名】 特許願
【整理番号】 0351724
【提出日】 平成15年 9月19日
【あて先】 特許庁長官殿
【国際特許分類】 G06F 9/06
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
 内
 【氏名】 櫻井 英樹
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
 内
 【氏名】 右田 和彦
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
 内
 【氏名】 下島 和樹
【発明者】
 【住所又は居所】 神奈川県川崎市中原区上小田中4丁目1番1号 富士通株式会社
 内
 【氏名】 長浜 栄一
【特許出願人】
 【識別番号】 000005223
 【氏名又は名称】 富士通株式会社
【代理人】
 【識別番号】 100074099
 【住所又は居所】 東京都千代田区二番町8番地20 二番町ビル3F
 【弁理士】
 【氏名又は名称】 大菅 義之
 【電話番号】 03-3238-0031
【選任した代理人】
 【識別番号】 100067987
 【住所又は居所】 神奈川県横浜市鶴見区北寺尾7-25-28-503
 【弁理士】
 【氏名又は名称】 久木元 彰
 【電話番号】 045-545-9280
【手数料の表示】
 【予納台帳番号】 012542
 【納付金額】 21,000円
【提出物件の目録】
 【物件名】 特許請求の範囲 1
 【物件名】 明細書 1
 【物件名】 図面 1
 【物件名】 要約書 1
 【包括委任状番号】 9705047

【書類名】 特許請求の範囲**【請求項 1】**

ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用する修正適用装置であって、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算する計算手段と、

前記顧客により指定された、前記指標に対する制約条件を格納する格納手段と、

計算された指標が前記制約条件を満たすか否かをチェックし、該計算された指標が該制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする適用手段と
を備えることを特徴とする修正適用装置。

【請求項 2】

ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用するコンピュータのためのプログラムであって、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

計算された指標が前記制約条件を満たすか否かをチェックし、

前記計算された指標が前記制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする
処理を前記コンピュータに実行させることを特徴とするプログラム。

【請求項 3】

いずれの顧客にも割り当てられていない空きサーバ上に、前記顧客が使用しているソフトウェアが稼動している処理対象サーバと同じソフトウェア構成を用意し、該空きサーバ上で該顧客が使用しているソフトウェアに前記修正情報を適用し、該処理対象サーバの代わりに該空きサーバを該顧客に割り当てることで、該ソフトウェアをアップデートする処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 4】

いずれの顧客にも割り当てられていない空きサーバ上で、いずれの顧客にも割り当てられていないソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートし、該ソフトウェアを必要とする顧客が現れたとき該空きサーバを該顧客に割り当てる処理を前記コンピュータに実行させることを特徴とする請求項 2 記載のプログラム。

【請求項 5】

ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用する修正適用方法であって、

計算手段が、前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

適用手段が、前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

前記適用手段が、計算された指標が前記制約条件を満たすか否かをチェックし、該計算された指標が該制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする
ことを特徴とする修正適用方法。

【書類名】明細書

【発明の名称】ソフトウェアに修正情報を適用する装置および方法

【技術分野】

【0001】

本発明は、ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいて、ソフトウェアに修正情報を適用する装置および方法に関する。

【背景技術】

【0002】

近年、ネットワーク環境を有しコンピュータシステムの維持管理を行うデータセンタにおいて複数のコンピュータリソースを複数の顧客に動的に割り当てることで、リソースの有効活用を図るシステムが提案されている。このようなシステムとしては、例えば、IBM社 (International Business Machines Corporation) のAutonomic Computing や、HP社 (Hewlett-Packard Company) のUtility Data Center、サン・マイクロシステムズ社 (Sun Microsystems, Inc.) のN1等が挙げられる。

【0003】

リソースの種類としては、CPU (中央処理装置)、ネットワーク装置等のハードウェアと、オペレーティングシステム (OS)、アプリケーションプログラム (以下、アプリケーションと記す) 等のソフトウェアがある。ソフトウェアには、OSやアプリケーション等のプログラムの他に、プログラムにより使用されるデータも含まれる。

【0004】

図23は、このようなシステムにおいて、各サーバで動作している各顧客の各ソフトウェアについて、どのバージョンのものが稼動しているかを示している。ここでは、サーバS1～S3が顧客Aに割り当てられており、サーバS5およびS6が顧客Bに割り当てられており、サーバS7が顧客Cに割り当てられている。

【0005】

例えば、サーバS1では顧客Aのシステムとして、オペレーティングシステムOS1とアプリケーションAP1が稼動しており、サーバS6では顧客Bのシステムとして、オペレーティングシステムOS2とデータベースDB2が稼動している。また、サーバS7では顧客Cのシステムとして、オペレーティングシステムOS1とデータベースDB2が稼動している。サーバS4では何も動作しておらず、このサーバは空き状態となっている。状況によっては、サーバS4でいずれかの顧客のシステムが稼動することもあり得る。

【0006】

このように、複数のサーバを複数の顧客が共有し動的に稼動状態が変更されるシステムは新しいシステムであり、リソースの保守運用についても新しい方法が求められる。特に、ソフトウェアについては、ベンダから頻繁に修正情報 (パッチ) が提供されるので、システムの維持管理のためにはそれらのパッチを随時システムに適用して保守していかなければならない。

【0007】

従来の保守運用方法では、データセンタのオペレータが人手で保守作業を行っていた。例えば、システム上のリソース情報を表形式で管理し、リソースの修正情報がベンダから提供された場合は、オペレータがその情報を分析して、ベンダにより指定されたシステム構成と現状のシステム構成との比較検討を行うことで、修正情報適用の可否を決めていた。実際に修正情報をリソースに適用する際には、あらかじめ顧客との間で取り決められたタイミング (例えば、週2回深夜1時～3時等) で顧客システムを停止して、人手で修正作業を行うケースがほとんどである。

【0008】

また、単一の顧客システムに対するプログラム修正の適用を支援するシステムも提案されている (例えば、特許文献1参照)。

【特許文献1】特開2000-10766号公報

【発明の開示】

【発明が解決しようとする課題】**【0009】**

しかしながら、上述した従来の保守運用方法には、次のような問題がある。

顧客サービスとして既に割り当てられ稼動しているリソースに対しては、そのままの状態ではパッチを適用することができず、これを適用するためにシステムを停止せざるを得なくなるという危険性がある。

【0010】

また、リソースが大量になると、どのソフトウェアがどのバージョンになっているのかを簡単に知ることができない。あるアプリケーションに対して緊急のパッチが発行されたとしても、それをどのソフトウェアに適用する必要があるのかを調べるのに手間がかかる。つまり、大量にあるリソースのそれぞれについて、どの程度新しいバージョンのアプリケーションが動作していてどの程度安全であるのかについて、統一的に知る手段がない。

【0011】

このため、どのサーバでどのソフトウェアが動作しているかを一元管理できて、ソフトウェアのアップデートまで自動化可能なシステムは、まだ実現されていない。しかし、サーバの数が大量になってくるとオペレータが管理可能な限界を超えてしまうので、システム自身が自動的に管理する機能は今後必ず必要になると考えられる。

【0012】

さらに、ソフトウェアに対するパッチの適用は必ずしもすべて必須というわけではない。また、顧客システムのカスタマイズ内容等によっては、最新のパッチを適用することによって不都合が発生することもあると考えられる。したがって、パッチ適用の可否を適切に判断する機能も望まれる。

【0013】

本発明の課題は、複数のソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいて、ソフトウェアに対して修正情報を的確に適用する装置および方法を提供することである。

【課題を解決するための手段】**【0014】**

図1は、本発明の修正適用装置の原理図である。図1の修正適用装置は、計算手段101、格納手段102、および適用手段103を備え、ソフトウェアを複数のサーバ104-1, 104-2, . . . , 104-mに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用する。

【0015】

計算手段101は、修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェア105の新しさとソフトウェア105のセキュリティレベルを統合した指標を計算する。格納手段102は、顧客により指定された、指標に対する制約条件105を格納する。適用手段103は、計算された指標が制約条件106を満たすか否かをチェックし、計算された指標が制約条件106を満たさなければ、ソフトウェア105に修正情報を適用してソフトウェア105をアップデートする。

【0016】

このような修正適用装置によれば、ソフトウェアの新しさとセキュリティレベルを統合した指標と、顧客毎のカスタマイズレベルを表す制約条件とを用いて、システム上で稼動している各顧客のソフトウェアが統一的に管理される。この指標は、新たな修正情報が発行される度に再計算され、あらかじめ設定された制約条件を満たさなくなると、発行された修正情報が該当する顧客のソフトウェアに適用される。

【0017】

このとき、適用手段103は、いずれの顧客にも割り当てられていない空きサーバ上に、顧客が使用しているソフトウェア105が稼動している処理対象サーバ104-2と同じソフトウェア構成を用意し、空きサーバ上でソフトウェア105に修正情報を適用し、処理対象サーバ104-2の代わりに空きサーバを顧客に割り当てることで、ソフトウェ

ア105をアップデートすることができる。

【0018】

また、適用手段103は、空きサーバ上で、いずれの顧客にも割り当てられていないソフトウェアに修正情報を適用してそのソフトウェアをアップデートし、そのソフトウェアを必要とする顧客が現れたとき空きサーバをその顧客に割り当てることもできる。

【0019】

計算手段101、格納手段102、および適用手段103は、例えば、後述する図2のパッチデータ管理部205、顧客使用ソフトウェア一覧表212、および動的アサイン管理部206に対応する。また、計算手段101が計算する指標は、後述するクリーンポイントに対応する。

【発明の効果】

【0020】

本発明によれば、各顧客システムの稼動状況やカスタマイズレベルに応じて、修正情報適用の可否が的確に判断される。これにより、顧客システムの安定稼動が可能になり、運用オペレータの作業が軽減される。また、空きサーバを利用してソフトウェアをアップデートすることにより、顧客システムを停止せずにリソースを最新かつ安全な状態にすることができる。

【発明を実施するための最良の形態】

【0021】

以下、図面を参照しながら、本発明を実施するための最良の形態を詳細に説明する。

一般に、顧客は、システムが一定のセキュリティレベルを維持することを期待するものである。そこで、本実施形態では、このセキュリティレベルを数値で表し、パッチの適用が必要か否かをシステムが自動的に判断できるようにするために、ソフトウェアリソースのクリーンポイントという指標を導入する。

【0022】

クリーンポイントは、ソフトウェアのクリーンさの程度を表す。ソフトウェアのバージョンが新しいほど、また、ソフトウェアのセキュリティレベルが高いほど、クリーンポイントの数値は大きくなる。

【0023】

本実施形態の修正適用装置は、以下のような機能を有する。

- ・各ソフトウェアについてクリーンポイントを計算する。
- ・各顧客が必要とするソフトウェアとそのソフトウェアに対する制約条件を管理するための顧客使用ソフトウェア一覧表を作成する。
- ・各顧客に対するリソースの割り当て状況を管理するための動的アサイン表を作成する。
- ・リソースとパッチの適用状態の組み合わせを管理するためのアップデート適用表を作成する。
- ・ソフトウェアベンダから提供されるパッチ情報をシステムで一元管理できる状態に正規化するためのパッチ情報解釈部を有する。
- ・パッチ情報を管理し、すべてのソフトウェアについてのクリーンポイントをあらかじめ計算するためのパッチデータ管理部を有する。
- ・システム上の全サーバについてどのソフトウェアが稼動しているかを一元管理するための動的アサイン管理部を有する。
- ・ある時点でどの顧客にも割り当てられていない空きサーバを利用して、あらかじめソフトウェアにアップデートパッチを適用しておき、そのソフトウェアを顧客に割り当てることで、即座にクリーンなソフトウェアを提供する。
- ・空きサーバを利用してソフトウェアに緊急アップデートパッチを適用し、空きサーバを順番に稼動状態に切り替えることで、稼動中のシステムをクリーンな状態に移行させる。
- ・各顧客が現有するすべてのソフトウェアについてクリーンポイントを計算し、顧客毎のシステムの安全性を可視化する。

【0024】

図2は、このような修正適用装置の構成図である。図2の修正適用装置は、パッチ情報解釈部203、204、パッチデータ管理部205、および動的アサイン管理部206を備え、共用サーバプール207上で動作している各顧客システムのソフトウェアに対して、パッチ提供元201および202から提供されたパッチを適用する。

【0025】

パッチ提供元201および202は、各ソフトウェアを作成するベンダであり、ベンダ毎にパッチ情報およびパッチデータの提供方法が異なる。パッチ情報は、パッチ提供元から提供されるパッチの適用方法に関する情報であり、パッチデータは、ソフトウェアに適用されるパッチ本体のデータである。パッチ情報解釈部203および204は、ベンダ毎に異なるパッチ提供方法の差異を吸収し、ソフトウェアの名称、バージョン、および重要度を正規化する。

【0026】

図2では、2つのパッチ提供元に応じて2つのパッチ情報解釈部が示されているが、3つ以上のパッチ提供元が存在する場合は、パッチ提供元の数に応じて複数のパッチ情報解釈部が用意される。また、1つのパッチ情報解釈部が複数のパッチ提供元から提供されるパッチ情報を解釈することも可能である。

【0027】

パッチデータ管理部205は、共用サーバプール207で使用される全ソフトウェアについて、データベース211に格納されたパッチ情報とパッチデータを管理し、パッチの時系列と重要度に基づいてソフトウェアのクリーンポイントを計算する。また、新しいパッチが発行された場合は、そのパッチ情報を動的アサイン管理部206に通知する。

【0028】

動的アサイン管理部206は、顧客使用ソフトウェア一覧表212、動的アサイン表213、およびアップデート適用表214を保持し、これらの表の内容を動的に管理する。そして、顧客使用ソフトウェア一覧表212を参照して新しいパッチの適用可否を判断し、適用可と判断したパッチを共用サーバプール207上のソフトウェアに適用する。適用処理の詳細については後述することにする。

【0029】

共用サーバプール207は複数のサーバS1～S7からなり、そのうち1つ以上のサーバ上で各顧客のシステムが動作する。状況に応じて稼動状態は異なり、ある時点で顧客Aに割り当てられていたリソースが、別の時点では顧客Bに割り当てられることもあり得る。

【0030】

次に、図3から図20までを参照しながら、図2の修正適用装置の処理についてより詳細に説明する。

図3は、図2のパッチ情報解釈部203により行われるパッチ情報解釈処理を示している。パッチ提供元201から発行されたパッチ情報301は、電子メール等の方法でパッチ情報解釈部203に転送される。パッチ情報解釈部203は、パッチ情報解釈対応表302を参照してパッチ情報301を解釈し、パッチ情報301から日付、ソフトウェア名、重要度、およびバージョンを含むパッチ管理情報303を抽出し、パッチ情報301に含まれるリンク先からパッチデータ304を取得する。そして、得られたパッチ管理情報303およびパッチデータ304をデータベース211に登録する。

【0031】

図4は、パッチ情報301、パッチ情報解釈対応表302、パッチ管理情報303、およびパッチデータ304の例を示している。パッチ情報301には、発行日401、影響ソフトウェア名402、セキュリティレベル403、およびダウンロード情報404が含まれている。また、パッチ情報解釈対応表302は、日付405、ソフトウェア名406、重要度407、バージョン408、およびパッチデータ409のエントリを有し、各エントリには必要な管理情報の記載位置が記録されている。

【0032】

パッチ情報解釈部 203 は、パッチ情報解釈対応表 302 の各エントリに記録された記載位置の情報に基づいて、パッチ情報 301 の対応する位置から管理情報を抽出する。抽出されたバージョンは、パッチのバージョンであると同時に、ソフトウェアのバージョンでもある。

【0033】

パッチ情報解釈対応表 302 の重要度 407 のエントリには、パッチ情報 301 に記載されたセキュリティレベル 403 の種類に応じて、レベルの低い方から順番に 1 から始まる番号で重要度が指定されている。ここでは、セキュリティレベル 403 に“緊急”と記載されているので、対応する重要度“4”がパッチ管理情報 303 として出力される。

【0034】

また、パッチ情報解釈部 203 は、パッチデータ 409 のエントリに記録された記載位置の情報に基づいて、パッチ情報 301 の対応する位置からリンク先の情報を抽出し、そのリンク先からパッチデータ 304 をダウンロードする。

【0035】

顧客が使用するソフトウェアのパッチ提供元毎にこのようなパッチ情報解釈対応表 302 を用意しておくことで、修正適用装置は、自動的に必要な情報を取得することができるようになる。

【0036】

図 5 は、上述したパッチ情報解釈処理のフローチャートである。パッチ情報解釈部 203 は、パッチ提供元 201 から新しいパッチ情報 301 を受け取ると、まず、パッチ情報解釈対応表 302 を参照して、パッチの適用対象となるソフトウェアの名称を抽出し（ステップ 501）、システムで使用しているソフトウェアか否かをチェックする（ステップ 502）。ソフトウェア名が使用ソフトウェアに該当しなければ、そのパッチを適用する必要がないので、そのまま処理を終了する。

【0037】

ソフトウェア名が使用ソフトウェアに該当すれば、次に、パッチの発行日を抽出し（ステップ 503）、パッチの重要度を抽出し（ステップ 504）、パッチのバージョンを抽出する（ステップ 505）。

【0038】

次に、パッチのダウンロード方法を抽出し（ステップ 506）、その方法に従ってパッチデータ 304 をダウンロードする（ステップ 507）。

そして、抽出された日付、ソフトウェア名、重要度、およびバージョンからなるパッチ管理情報 303 と、ダウンロードされたパッチデータ 304 とをデータベース 211 に登録して（ステップ 508）、処理を終了する。図 2 のパッチ情報解釈部 204 も、基本的にパッチ情報解釈部 203 と同様の処理を行う。

【0039】

図 6 は、図 2 のパッチデータ管理部 205 により用いられるクリーンポイントの計算方法の例を示している。パッチデータ管理部 205 は、データベース 211 に登録されたソフトウェア A の各バージョンのパッチ管理情報に基づいて、各バージョンのクリーンポイントを計算する。このとき、まず、各バージョンの基本ポイントと重要度から、各バージョンにおける正規化前のクリーンポイントを計算し、得られた値を最新バージョンの値をもとにして正規化する。

【0040】

この計算方法では、まず、ソフトウェア A の初版であるバージョン 1.0 の発行日と、最新バージョンであるバージョン 1.3 の発行日をもとに、時間軸が作成される。次に、初版をポイント“1”、最新バージョンをポイント“100”として、単純に各バージョンの発行日に対応する基本ポイントが計算され、基本ポイントグラフが作成される。

【0041】

図 6 の例では、ソフトウェア A の各バージョンの基本ポイントは、以下のように算出される。

バージョン 1. 0 基本ポイント “1”
 バージョン 1. 1 基本ポイント “80”
 バージョン 1. 2 基本ポイント “95”
 バージョン 1. 3 基本ポイント “100”

次に、初版の重要度を“1”として、各バージョンの重要度をもとにした重要度グラフが作成される。最後に、各バージョンの重要度に応じて基本ポイントグラフの傾きを変更したグラフが作成され、最新バージョンのポイントが“100”となるように、各バージョンのポイントが正規化される。こうして算出された各バージョンのポイントが、クリーンポイントとして記録される。

【0042】

以上のようなクリーンポイントの計算方法は、次式により表される。

$$a_n = a_{n-1} + (t_n - t_{n-1}) \times s_n \quad (n=1, \dots, N) \quad (1)$$

$$C_1 = 1 \quad (2)$$

$$C_i = (a_i / a_N) \times 100 \quad (i=2, \dots, N) \quad (3)$$

ただし、(1)式において、 t_n は n 番目のバージョンの基本ポイント、 s_n は n 番目のバージョンの重要度、 a_n は n 番目のバージョンの正規化される前のクリーンポイントを表し、 $t_0=0$ 、 $a_0=0$ とする。1番目のバージョンは初版に相当し、 N 番目のバージョンは最新バージョンに相当する。また、(2)および(3)式において、 C_i ($i=1, \dots, N$)は i 番目のバージョンの正規化されたクリーンポイントを表す。

【0043】

実際には、パッチデータ管理部205は、図6に示したようなグラフを画面上に表示するのではなく、(1)～(3)式を用いてクリーンポイントを直接算出する。この計算方法によれば、古いけれど重要なパッチや新しいけれどそれほど重要ではないパッチも含めて、すべてのバージョンのパッチを統一的に数値で表現することが可能になる。

【0044】

なお、(1)～(3)式はクリーンポイントの計算方法の一例を示しているに過ぎず、クリーンポイントの計算方法はこれに限定されるものではない。

図7は、図2の動的アサイン管理部206が保持する顧客使用ソフトウェア一覧表212の例を示している。図7の顧客使用ソフトウェア一覧表は、それぞれの顧客が使用するソフトウェアの一覧を管理するために用いられ、顧客名、ソフトウェア名、バージョン束縛条件、およびCL (クリーンポイント) 閾値の各欄からなる。

【0045】

バージョン束縛条件の欄には、ソフトウェアを所定のバージョンで動作させる必要がある場合にそのバージョンを指定する条件が記録され、CL制約条件の欄には、各顧客により指定された、ソフトウェアのクリーンポイントが満たすべき制約条件が記録される。この制約条件には、例えば、クリーンポイントの閾値が含まれる。

【0046】

図7の701～703の各エントリは、以下のような意味を有する。

701: 顧客“USER1”は、ソフトウェア“Software1”の様々なバージョンのうち、クリーンポイントが85以上のものを使用する。

702: 顧客“USER1”は、ソフトウェア“Software2”の様々なバージョンのうち、クリーンポイントが90以上のものを使用する。“Software2”は頻繁にバージョンアップされるので、“Software1”よりクリーンポイントの閾値が高く設定されている。

703: 顧客“USER1”は、ソフトウェア“Software3”のバージョン1.3を使用する。ただし、この場合、バージョン1.3のクリーンポイントは“70”であ

るものとする。

【0047】

動的アサイン管理部206は、このような顧客使用ソフトウェア一覧表を用いて顧客が要求しているソフトウェアとクリーンレベルとを認識する。

図8は、図2の動的アサイン管理部206が保持する動的アサイン表213の例を示している。図8の動的アサイン表は、現在のシステムにおいてどのサーバ上でどの顧客に対してどのソフトウェアが割り当てられているかの一覧を管理するために用いられ、サーバ名、顧客名、インストールソフトウェア名、およびバージョンの各欄からなる。インストールソフトウェア名の欄には、サーバにインストールされたソフトウェアの名称が記録され、バージョンの欄には、そのソフトウェアのバージョンが記録される。

【0048】

図8の801～804の各エントリは、以下のような意味を有する。

801:サーバ“S1”上で顧客“USER1”のために、ソフトウェア“Software1”のバージョン“1.3”が動作している。

802:サーバ“S1”上で顧客“USER1”のために、ソフトウェア“Software2”のバージョン“5.2”が動作している。

803:サーバ“S2”上でソフトウェア“Software1”のバージョン“1.3”が動作しているが、現在どの顧客にも割り当てられていない。

804:サーバ“S3”上で顧客“USER2”のために、ソフトウェア“Software3”のバージョン“3.3”が動作している。

【0049】

動的アサイン管理部206は、このような動的アサイン表を用いて現在サーバ上で稼動している全顧客システムの状態を認識する。

図9は、図2の動的アサイン管理部206が保持するアップデート適用表214の例を示している。図9のアップデート適用表は、ソフトウェアのどのバージョンがどのサーバ上で動作しているかについてシステムの現状を管理するために用いられ、ソフトウェア名、サーバ名、バージョン、およびCLの各欄からなる。

【0050】

図9の901～904の各エントリは、以下のような意味を有する。

901:ソフトウェア“Software1”のバージョン“1.3”（クリーンポイント“70”）がサーバ“S1”上で動作している。

902:ソフトウェア“Software1”のバージョン“1.4”（クリーンポイント“80”）がサーバ“S2”上で動作している。

903:ソフトウェア“Software1”のバージョン“1.5”（クリーンポイント“100”）がサーバ“S3”上で動作している。

904:ソフトウェア“Software2”のバージョン“5.3”（クリーンポイント“89”）がサーバ“S2”上で動作している。

【0051】

動的アサイン管理部206は、このようなアップデート適用表を用いて全サーバ上で動作しているソフトウェアの状態を認識する。このアップデート適用表を図8の動的アサイン表と組み合わせて参照すれば、顧客に割り当てられていない空きサーバを利用してソフトウェアの自動アップデートを行うことができる。

【0052】

図10は、図2の修正適用装置により行われる修正適用処理のフローチャートである。パッチ提供元から新しいパッチ情報を受け取ると、まず、パッチ情報解釈部は、図5に示したパッチ情報解釈処理を行って、パッチ管理情報およびパッチデータをデータベース211に登録する（ステップ1001）。

【0053】

次に、パッチデータ管理部205は、データベース211に格納されたパッチ管理情報を参照して、パッチの適用対象となるソフトウェアの各バージョンのクリーンポイントを

計算し、得られたクリーンポイントの値を動的アサイン管理部 206 に通知する（ステップ 1002）。

【0054】

動的アサイン管理部 206 は、パッチ管理情報のソフトウェア名をキーとして顧客使用ソフトウェア一覧表 212 を検索し、そのソフトウェアに対して各顧客が設定している CL 制約条件を取得する。また、そのソフトウェア名をキーとして動的アサイン表 213 を検索し、各顧客に割り当てられているそのソフトウェアのバージョンを取得する。

【0055】

そして、通知された各バージョンのクリーンポイントのうち、各顧客のバージョンに対応するクリーンポイントの値がその顧客の CL 制約条件を満たすか否かをチェックする（ステップ 1003）。各バージョンのクリーンポイントの値が対応する顧客の CL 制約条件を満たしている場合は、パッチを適用する必要があると判断し、そのまま処理を終了する。

【0056】

いずれかのバージョンに対応するクリーンポイントの値が対応する顧客の CL 制約条件を満たさない場合は、その顧客のソフトウェアに対してパッチを適用する必要があると判断する。そこで、共用サーバプール 207 の空きサーバを利用して、その顧客に割り当てられているサーバと同じソフトウェア構成で、かつ、対象ソフトウェアにパッチが適用された状態を作り出す（ステップ 1004～1010）。このとき、空きサーバ上に必要なソフトウェアがそろっていない場合は、状況に応じて各ソフトウェアのインストール、アップデート、およびアンインストールが行われる。

【0057】

動的アサイン管理部 206 は、まず、動的アサイン表 213 を参照して、対象ソフトウェアがインストールされているサーバと空きサーバを検出し、前者を処理対象サーバとする。そして、処理対象サーバと同じ構成を実現するために不要なソフトウェアが空きサーバにあるか否かをチェックし（ステップ 1004）、そのようなソフトウェアがあれば、そのソフトウェアのアンインストールを実行する（ステップ 1005）。

【0058】

次に、動的アサイン表 213 を参照して、対象ソフトウェア以外のソフトウェアであっても、処理対象サーバのものより古いバージョンのソフトウェアが空きサーバにあるか否かをチェックする（ステップ 1006）。そして、そのようなソフトウェアがあれば、そのソフトウェアをアップデートする（ステップ 1007）。

【0059】

次に、動的アサイン表 213 を参照して、処理対象サーバと同じ構成を実現するために必要なソフトウェアがすべて空きサーバにあるか否かをチェックし（ステップ 1008）、不足しているソフトウェアがあれば、そのソフトウェアを空きサーバにインストールする（ステップ 1009）。

【0060】

そして、空きサーバ上で対象ソフトウェアにパッチを適用してそのソフトウェアをアップデートし（ステップ 1010）、動的アサイン表 213 を書き換えて、顧客へのサーバの割り当てを処理対象サーバから空きサーバへ変更する（ステップ 1011）。

【0061】

次に、動的アサイン表 213 を参照して、同じ顧客に割り当てられた残りのサーバの中にパッチを適用する必要があるサーバがあるか否かをチェックし（ステップ 1012）、そのようなサーバがあれば、それを処理対象サーバとしてステップ 1004 以降の処理を繰り返す。

【0062】

そして、その顧客のすべてのサーバについて処理が終了すると、次に、動的アサイン表 213 を参照して、残りの顧客に割り当てられたサーバにパッチを適用する必要があるか否かをチェックする（ステップ 1013）。そのようなサーバがあればステップ 1004

以降の処理を繰り返し、すべての顧客のシステムについて処理が終了すると、現状のシステム構成に合わせてアップデート適用表 214 を書き換えて修正適用処理を終了する。

【0063】

例えば、ある瞬間において、図 11 のような状態でシステムが稼動していたとする。この状態では、サーバ S1、S2、および S3 が顧客 A に割り当てられており、サーバ S5 および S6 が顧客 B に割り当てられており、サーバ S7 が顧客 C に割り当てられている。サーバ S4 は空きサーバである。

【0064】

サーバ S1 および S2 ではオペレーティングシステム OS1 とアプリケーション AP1 が稼動しており、サーバ S3 ではオペレーティングシステム OS1 とデータベース DB1 が稼動しており、サーバ S4 ではオペレーティングシステム OS1 のみが稼動している。また、サーバ S5 ではオペレーティングシステム OS2 のみが稼動しており、サーバ S6 ではオペレーティングシステム OS2 とアプリケーション AP1 が稼動しており、サーバ S7 ではオペレーティングシステム OS1 とデータベース DB2 が稼動している。オペレーティングシステム OS2 は、オペレーティングシステム OS1 の新しいバージョンに相当する。

【0065】

ここで、AP1 に対してセキュリティホール対応等の緊急アップデートパッチが発行された場合、以下のような手順で修正適用処理が行われる。

動的アサイン管理部 206 は、まず、サーバ S1 の AP1 にパッチを適用して AP2 にアップデートすることを決定する。サーバ S4 が空いているので、このサーバを利用して、サーバ S1 と同じソフトウェア構成で、かつ、AP1 が AP2 にアップデートされた状態を作り出すことにする。サーバ S4 には既に OS1 がインストールされているが AP1 がインストールされていないので、図 12 に示すように、サーバ S4 に AP1 をインストールする。

【0066】

次に、図 13 に示すように、サーバ S4 の AP1 にパッチを適用して AP2 にアップデートし、サーバ S1 の動的アサインをサーバ S4 に切り替える。これにより、図 14 に示すように、サーバ S4 は顧客 A に割り当てられてクリーンな稼動状態になり、サーバ S1 は空きサーバとなる。

【0067】

次に、動的アサイン管理部 206 は、サーバ S2 の AP1 を AP2 にアップデートすることを決定する。サーバ S1 が空いているので、このサーバを利用して、サーバ S2 と同じソフトウェア構成で、かつ、AP1 が AP2 にアップデートされた状態を作り出すことにする。サーバ S1 には既に OS1 と AP1 がインストールされているので、図 15 に示すように、サーバ S1 の AP1 を AP2 にアップデートし、サーバ S2 の動的アサインをサーバ S1 に切り替える。これにより、図 16 に示すように、サーバ S1 は再び顧客 A に割り当てられてクリーンな稼動状態になり、サーバ S2 は空きサーバとなる。

【0068】

次に、動的アサイン管理部 206 は、サーバ S6 の AP1 を AP2 にアップデートすることを決定する。サーバ S2 が空いているので、このサーバを利用して、サーバ S6 と同じソフトウェア構成で、かつ、AP1 が AP2 にアップデートされた状態を作り出すことにする。サーバ S2 には OS1 と AP1 がインストールされているが、必要なのは OS2 と AP2 がインストールされた状態である。そこで、図 17 に示すように、まず、OS1 を OS2 にアップデートし、次に、AP1 を AP2 にアップデートして、サーバ S6 の動的アサインをサーバ S2 に切り替える。これにより、図 18 に示すように、サーバ S2 は顧客 B に割り当てられてクリーンな稼動状態になり、サーバ S6 は空きサーバとなる。こうして、稼動中のすべての AP1 について AP2 へのアップデートが完了する。

【0069】

このような修正適用処理によれば、空きサーバを利用して緊急度の高い顧客システムか

ら玉突き式にパッチを適用することで、パッチ適用のための無駄なシステム停止を最小限に抑えることが可能になる。空きサーバを常に1つ以上確保しておく必要があるので一見無駄が発生するように見えるが、実際には、この空きサーバを修正適用装置の様々な処理に利用することが可能である。例えば、空きサーバの能力を利用してクリーンポイント計算処理や動的アサイン管理を行えば、このサーバを有効に利用することができる。

【0070】

図10の修正適用処理では、既に顧客に割り当てられているソフトウェアに対してパッチが適用されているが、空きサーバを利用すれば、まだ顧客に割り当てられていないソフトウェアに対しても事前にパッチを適用しておくことが可能である。これにより、実際にシステムが稼動している状態で、あらかじめ空きサーバにアップデートされたソフトウェアを用意しておくことができ、システムがクリーンな状態に保たれる。したがって、次に顧客への動的アサインが発生したときに、迅速にクリーンなりソースを割り当てることが可能になる。

【0071】

図19は、このような事前修正適用処理のフローチャートである。この処理は、例えば、図10の修正適用処理の後で引き続いて行われる。動的アサイン管理部206は、まず、動的アサイン表213を参照して空きサーバがあるか否かをチェックし（ステップ1901）、空きサーバがなければそのまま処理を終了する。

【0072】

空きサーバがあれば、動的アサイン表213を参照して、そのサーバにパッチの適用対象となるソフトウェアがあるか否かをチェックし（ステップ1902）、対象ソフトウェアがなければ、そのソフトウェアを空きサーバにインストールする（ステップ1903）。そして、空きサーバ上で対象ソフトウェアにパッチを適用してそのソフトウェアをアップデートする（ステップ1904）。

【0073】

次に、他の対象ソフトウェアがあるか否かをチェックし（ステップ1905）、そのようなソフトウェアがあれば、そのソフトウェアについてステップ1901以降の処理を繰り返す。そして、すべての対象ソフトウェアについて処理が終了すると、現状のシステム構成に合わせて動的アサイン表213およびアップデート適用表214を書き換えて、事前修正適用処理を終了する。

【0074】

例えば、サーバS1～S8からなる共用サーバプールにおいて、図20のような状態でシステムが稼動していたとする。この状態では、サーバS1、S2、およびS3が顧客Aに割り当てられており、サーバS5およびS6が顧客Bに割り当てられており、サーバS7が顧客Cに割り当てられている。サーバS4およびS8は空きサーバである。

【0075】

サーバS1およびS2ではオペレーティングシステムOS1とアプリケーションAP1が稼動しており、サーバS3ではオペレーティングシステムOS1とデータベースDB1が稼動しており、サーバS4ではオペレーティングシステムOS2とアプリケーションAP1が稼動している。また、サーバS5ではオペレーティングシステムOS2のみが稼動しており、サーバS6ではオペレーティングシステムOS2とデータベースDB2が稼動している。さらに、サーバS7ではオペレーティングシステムOS1とデータベースDB2が稼動しており、サーバS8ではオペレーティングシステムOS2のみが稼動している。データベースDB2は、データベースDB1の新しいバージョンに相当する。

【0076】

ここで、AP1およびDB2に対してアップデート用のパッチが発行された場合、以下のような手順で事前修正適用処理が行われる。

動的アサイン管理部206は、まず、空きサーバS4を検出する。サーバS4では既にOS2とAP1が稼動しているので、AP1にパッチを適用してAP2にアップデートする。

【0077】

次に、空きサーバS8を検出する。サーバS8ではOS2のみが稼動しているが、AP1に対するパッチは既にサーバS4で適用済みなので、サーバS8はDB2に対するパッチを適用するために利用することにする。そこで、まず、サーバS8にDB2をインストールし、次に、そのDB2にパッチを適用してDB3にアップデートする。

【0078】

こうして、共用サーバプールには、AP1およびDB2の最新バージョンであるAP2およびDB3の稼動状態が実現される。ここで仮に、OS2とAP2を必要とする顧客への動的アサインが発生すれば、即座にサーバS4をその顧客に割り当てることができ、OS2とDB3を必要とする顧客への動的アサインが発生すれば、即座にサーバS8をその顧客に割り当てることができる。

【0079】

ところで、図2の修正適用装置および各サーバは、例えば、図21に示すような情報処理装置（コンピュータ）を用いて構成することができる。図21の情報処理装置は、CPU2101、メモリ2102、入力装置2103、出力装置2104、外部記憶装置2105、媒体駆動装置2106、ネットワーク接続装置2107を備え、それらはバス2108により互いに接続されている。

【0080】

メモリ2102は、例えば、ROM（read only memory）、RAM（random access memory）等を含み、処理に用いられるプログラムおよびデータを格納する。CPU2101は、メモリ2102を利用してプログラムを実行することにより、必要な処理を行う。

【0081】

図2のパッチ情報解釈部203、204、パッチデータ管理部205、および動的アサイン管理部206は、メモリ2102に格納されたプログラムを実行することにより実現される機能に対応する。また、図2の顧客使用ソフトウェア一覧表212、動的アサイン表213、およびアップデート適用表214と図3のパッチ情報解釈対応表302は、メモリ2102に格納されたデータに対応する。

【0082】

入力装置2103は、例えば、キーボード、ポインティングデバイス、タッチパネル等であり、オペレータからの指示や情報の入力に用いられる。出力装置2104は、例えば、ディスプレイ、プリンタ、スピーカ等であり、オペレータへの問い合わせや処理結果の出力に用いられる。

【0083】

外部記憶装置2105は、例えば、磁気ディスク装置、光ディスク装置、光磁気ディスク装置、テープ装置等である。情報処理装置は、この外部記憶装置2105に、上記プログラムおよびデータを格納しておき、必要に応じて、それらをメモリ2102にロードして使用する。また、外部記憶装置2105は、図2のデータベース211としても使用される。

【0084】

媒体駆動装置2106は、可搬記録媒体2109を駆動し、その記録内容にアクセスする。可搬記録媒体2109は、メモ리카ード、フレキシブルディスク、CD-ROM（compact disk read only memory）、光ディスク、光磁気ディスク等の任意のコンピュータ読み取り可能な記録媒体である。オペレータは、この可搬記録媒体2109に上記プログラムおよびデータを格納しておき、必要に応じて、それらをメモリ2102にロードして使用する。

【0085】

ネットワーク接続装置2107は、LAN（local area network）等の任意の通信ネットワークに接続され、通信に伴うデータ変換を行う。情報処理装置は、必要に応じて、上記プログラムおよびデータを外部の装置からネットワーク接続装置2107を介して受け取り、それらをメモリ2102にロードして使用する。また、ネットワーク接続装置21

07は、パッチ情報やパッチデータを受信するためにも使用される。

【0086】

図22は、図21の情報処理装置にプログラムおよびデータを供給することのできるコンピュータ読み取り可能な記録媒体を示している。可搬記録媒体2109やサーバ2201のデータベース2203に格納されたプログラムおよびデータは、情報処理装置2202のメモリ2102にロードされる。サーバ2201は、そのプログラムおよびデータを搬送する搬送信号を生成し、ネットワーク上の任意の伝送媒体を介して情報処理装置2202に送信する。CPU2101は、そのデータを用いてそのプログラムを実行し、必要な処理を行う。

【0087】

(付記1) ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用する修正適用装置であって、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算する計算手段と、

前記顧客により指定された、前記指標に対する制約条件を格納する格納手段と、

計算された指標が前記制約条件を満たすか否かをチェックし、該計算された指標が該制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする適用手段と

を備えることを特徴とする修正適用装置。

【0088】

(付記2) ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用するコンピュータのためのプログラムであって、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

計算された指標が前記制約条件を満たすか否かをチェックし、

前記計算された指標が前記制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする処理を前記コンピュータに実行させることを特徴とするプログラム。

【0089】

(付記3) いずれの顧客にも割り当てられていない空きサーバ上に、前記顧客が使用しているソフトウェアが稼動している処理対象サーバと同じソフトウェア構成を用意し、該空きサーバ上で該顧客が使用しているソフトウェアに前記修正情報を適用し、該処理対象サーバの代わりに該空きサーバを該顧客に割り当てることで、該ソフトウェアをアップデートする処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0090】

(付記4) いずれの顧客にも割り当てられていない空きサーバ上で、いずれの顧客にも割り当てられていないソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートし、該ソフトウェアを必要とする顧客が現れたとき該空きサーバを該顧客に割り当てる処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0091】

(付記5) 前記修正情報の発行時に提供される情報から該修正情報の日付および重要度の情報を抽出し、抽出された情報を用いて前記指標を計算する処理を前記コンピュータに実行させることを特徴とする付記2記載のプログラム。

【0092】

(付記6) 前記修正情報が適用されるソフトウェアの複数のバージョンのうち前記顧客が使用しているバージョンについて前記指標を計算し、該顧客が使用しているバージョンの指標が前記制約条件を満たさなければ、該バージョンに該修正情報を適用する処理を

前記コンピュータに実行させることを特徴とする付記 2 記載のプログラム。

【0093】

(付記 7) ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用するコンピュータのためのプログラムを記録した記録媒体であって、

前記プログラムが、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

計算された指標が前記制約条件を満たすか否かをチェックし、

前記計算された指標が前記制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする
処理を前記コンピュータに実行させることを特徴とするコンピュータ読み取り可能な記録媒体。

【0094】

(付記 8) ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用するコンピュータのためのプログラムを搬送する搬送信号であって、

前記プログラムが、

前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

計算された指標が前記制約条件を満たすか否かをチェックし、

前記計算された指標が前記制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする
処理を前記コンピュータに実行させることを特徴とする搬送信号。

(付記 9) ソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいてソフトウェアに修正情報を適用する修正適用方法であって、

計算手段が、前記修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさと該ソフトウェアのセキュリティレベルを統合した指標を計算し、

適用手段が、前記顧客により指定された、前記指標に対する制約条件を格納手段から読み出し、

前記適用手段が、計算された指標が前記制約条件を満たすか否かをチェックし、該計算された指標が該制約条件を満たさなければ、前記顧客が使用しているソフトウェアに前記修正情報を適用して該ソフトウェアをアップデートする

ことを特徴とする修正適用方法。

【産業上の利用可能性】

【0095】

多数の顧客システムを取り扱うデータセンタに本発明を適用すれば、稼動中のサービスを停止させることなく、コンピュータリソースが自動的にクリーンな状態に保たれる。

【図面の簡単な説明】

【0096】

【図 1】本発明の修正適用装置の原理図である。

【図 2】修正適用装置の構成図である。

【図 3】パッチ情報解釈部の処理を示す図である。

【図 4】パッチ情報を示す図である。

【図 5】パッチ情報解釈処理のフローチャートである。

【図 6】クリーンポイントの計算方法を示す図である。

【図 7】顧客使用ソフトウェア一覧表を示す図である。

【図 8】動的アサイン表を示す図である。

【図 9】 アップデート適用表を示す図である。

【図 10】 修正適用処理のフローチャートである。

【図 11】 共用サーバプールの第 1 の状態を示す図である。

【図 12】 共用サーバプールの第 2 の状態を示す図である。

【図 13】 共用サーバプールの第 3 の状態を示す図である。

【図 14】 共用サーバプールの第 4 の状態を示す図である。

【図 15】 共用サーバプールの第 5 の状態を示す図である。

【図 16】 共用サーバプールの第 6 の状態を示す図である。

【図 17】 共用サーバプールの第 7 の状態を示す図である。

【図 18】 共用サーバプールの第 8 の状態を示す図である。

【図 19】 事前修正適用処理のフローチャートである。

【図 20】 事前修正適用処理を示す図である。

【図 21】 情報処理装置の構成図である。

【図 22】 記録媒体を示す図である。

【図 23】 サーバ共用システムを示す図である。

【符号の説明】

【0097】

101 計算手段

102 格納手段

103 適用手段

104-1、104-2、104-m サーバ

105 ソフトウェア

106 制約条件

201、202 パッチ提供元

203、204 パッチ情報解釈部

205 パッチデータ管理部

206 動的アサイン管理部

207 共用サーバプール

211 データベース

212 顧客使用ソフトウェア一覧表

213 動的アサイン表

214 アップデート適用表

301 パッチ情報

302 パッチ情報解釈対応表

303 パッチ管理情報

304、409 パッチデータ

401 発行日

402 影響ソフトウェア名

403 セキュリティレベル

404 ダウンロード情報

405 日付

406 ソフトウェア名

407 重要度

408 バージョン

701、702、703、801、802、803、804、901、902、903

、904 エントリ

2101 CPU

2102 メモリ

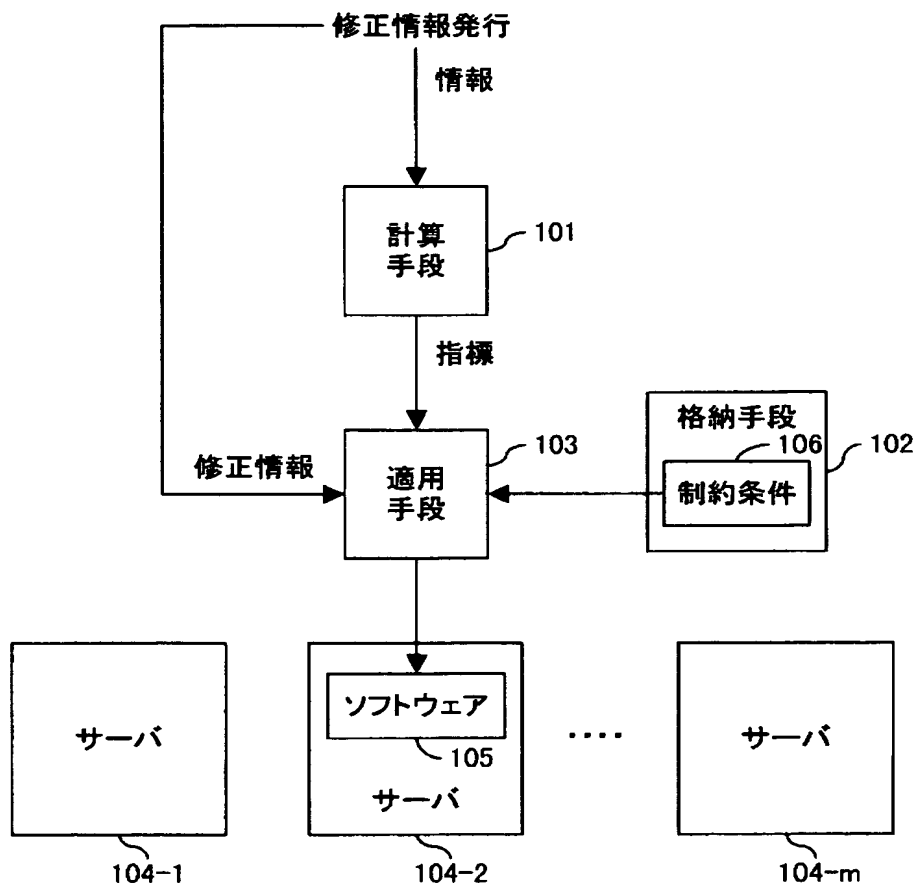
2103 入力装置

2104 出力装置

2 1 0 5 外部記憶装置
 2 1 0 6 媒体駆動装置
 2 1 0 7 ネットワーク接続装置
 2 1 0 8 バス
 2 1 0 9 可搬記録媒体
 2 2 0 1 サーバ
 2 2 0 2 情報処理装置
 2 2 0 3 データベース

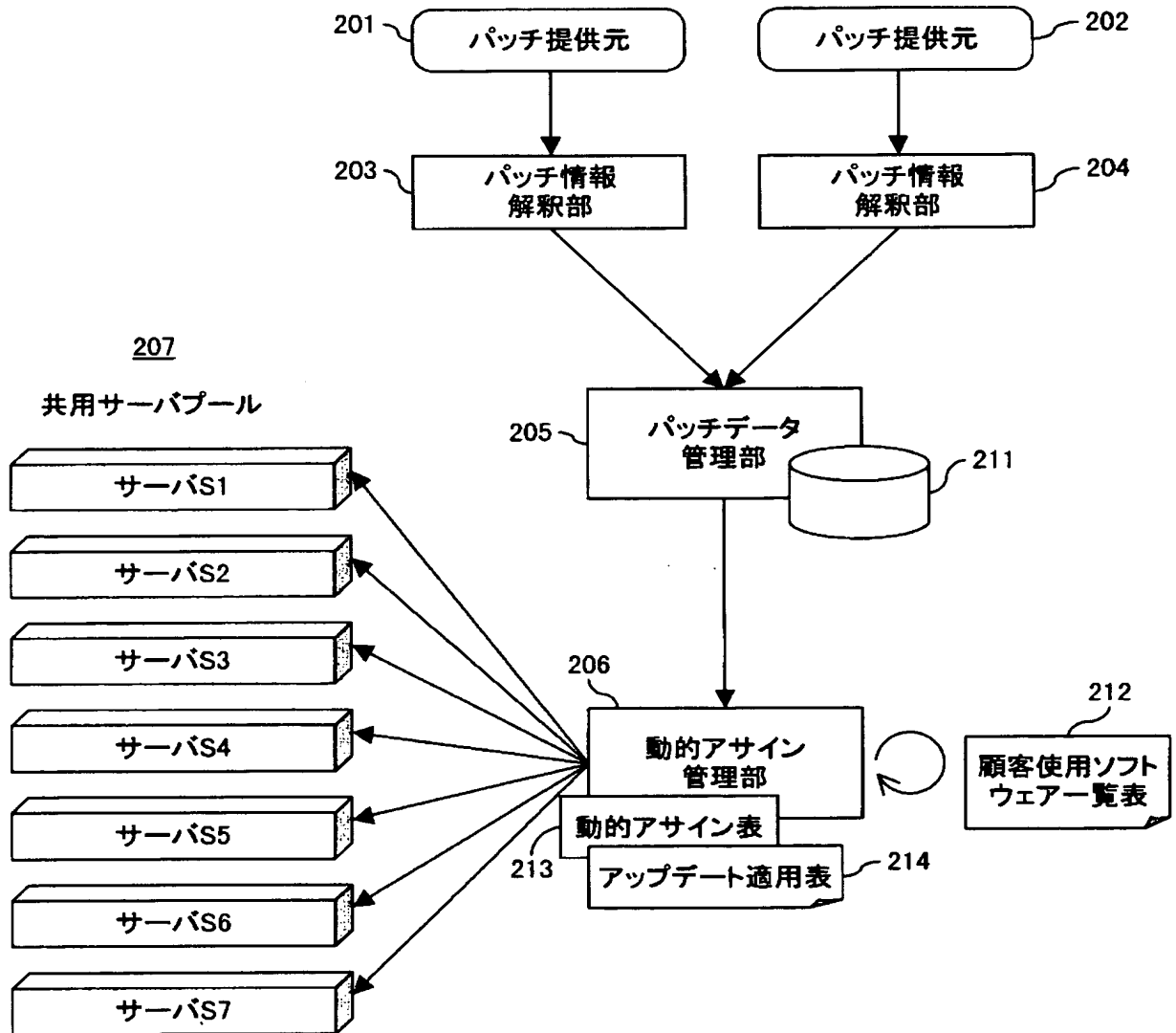
【書類名】 図面
【図 1】

本発明の修正適用装置の原理図



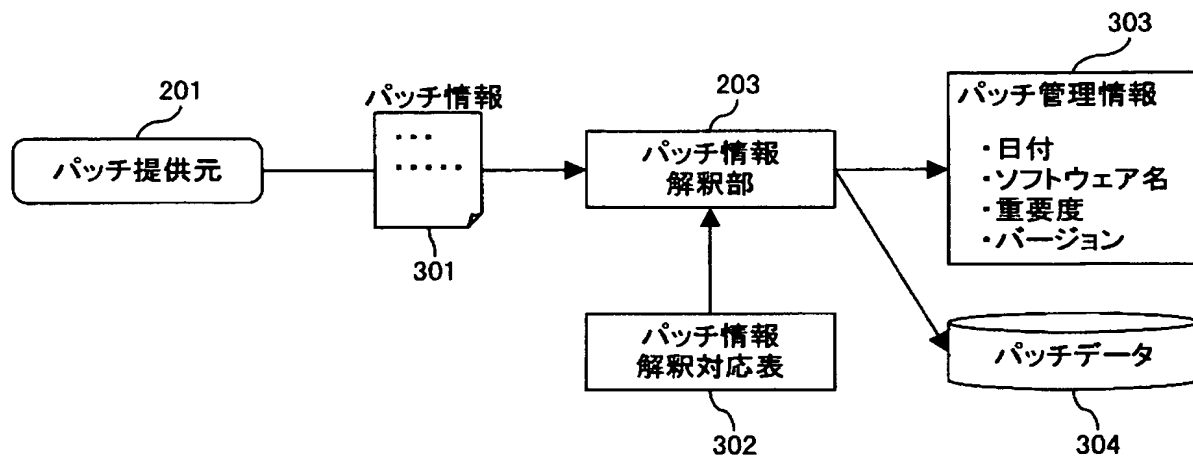
【図 2】

修正適用装置の構成図



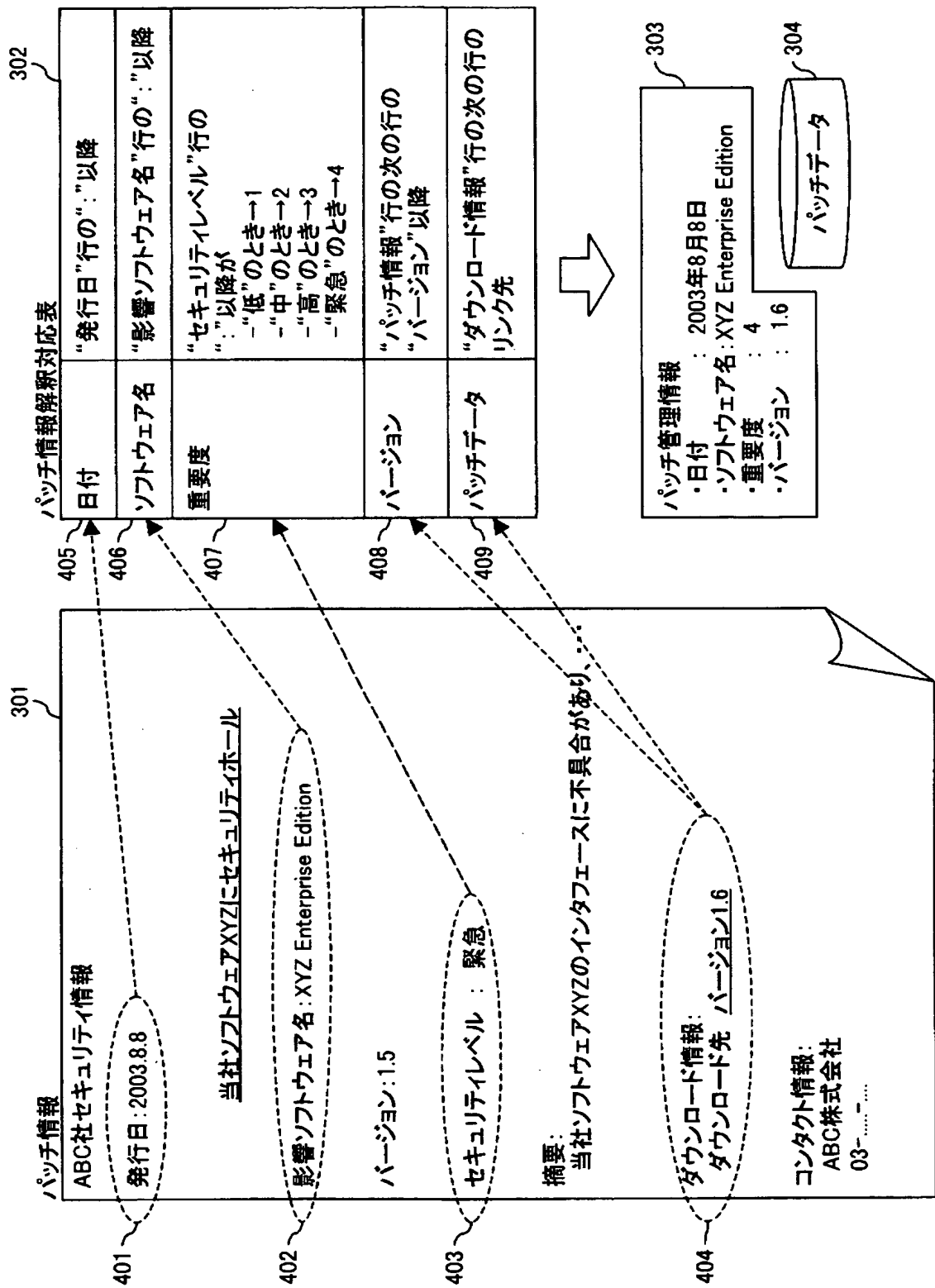
【図 3】

パッチ情報解釈部の処理を示す図



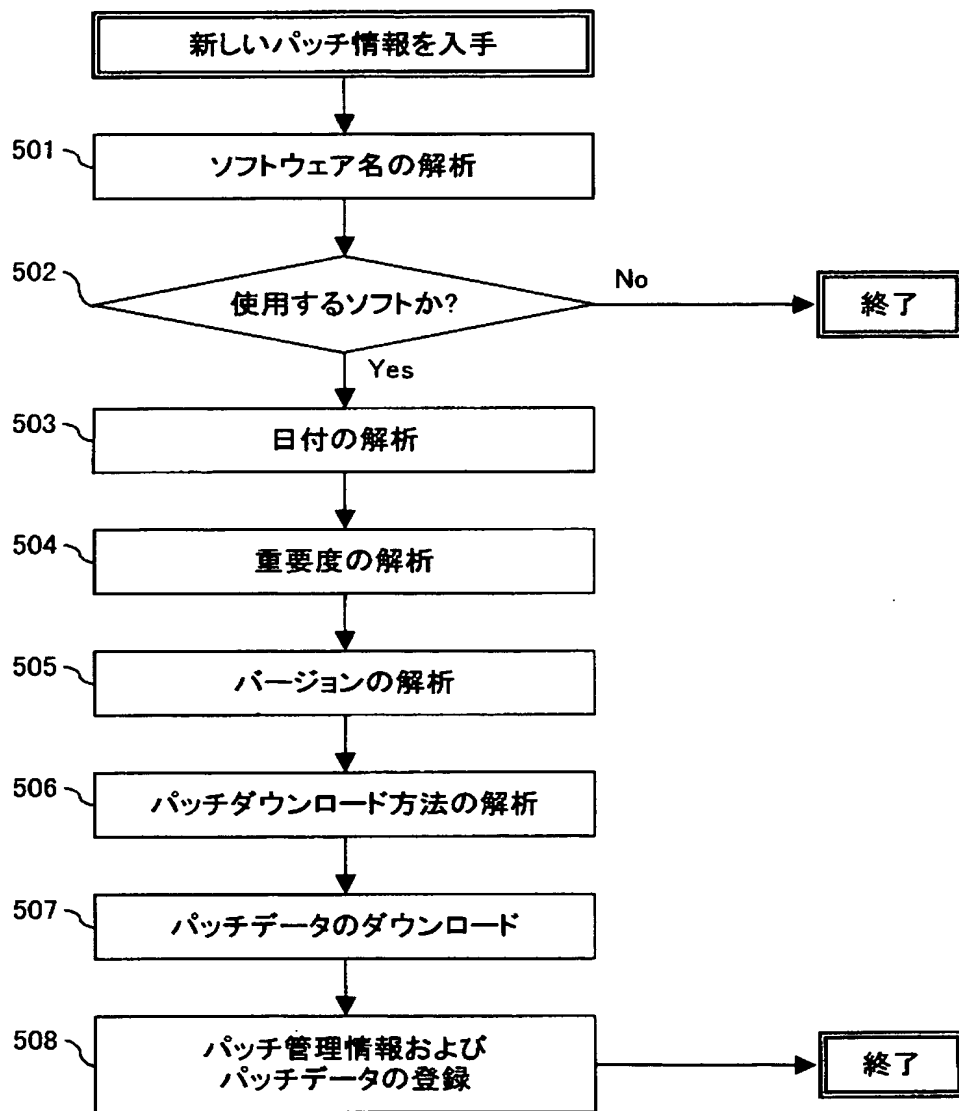
【図4】

パッチ情報を示す図



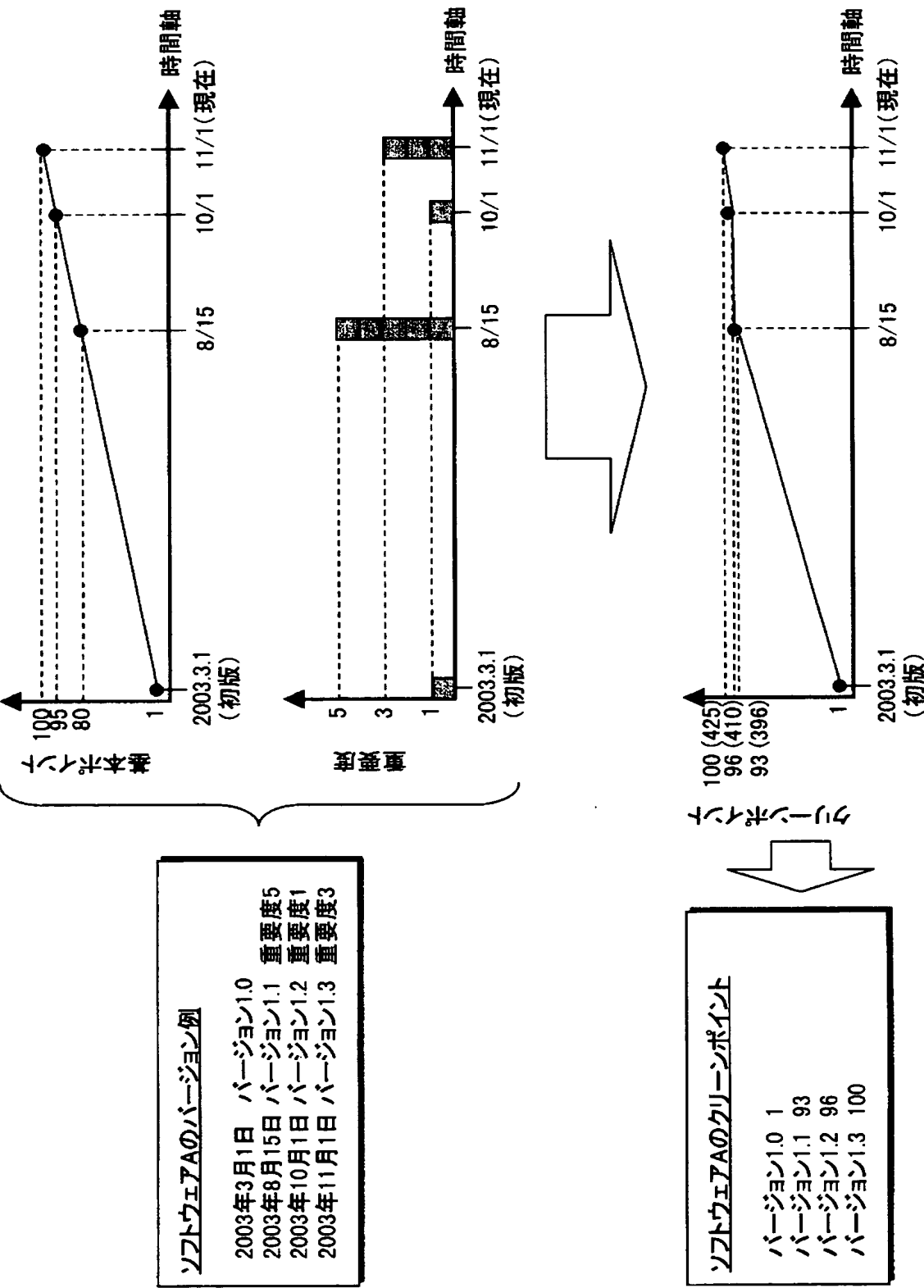
【図 5】

パッチ情報解釈処理のフローチャート



【図 6】

クリーンポイントの計算方法を示す図



【図 7】

顧客使用ソフトウェア一覧表を示す図

	顧客名	ソフトウェア名	バージョン束縛条件	CL制約条件
701	USER1	Software1	なし	85以上
702	USER1	Software2	なし	90以上
703	USER1	Software3	1.3	70以下
	USER2	...		

【図 8】

動的アサイン表を示す図

	サーバ名	顧客名	インストールソフトウェア名	バージョン
801	S1	USER1	Software1	1.3
802	S1	USER1	Software2	5.2
803	S2	なし	Software1	1.3
804	S3	USER2	Software3	3.3
	...			

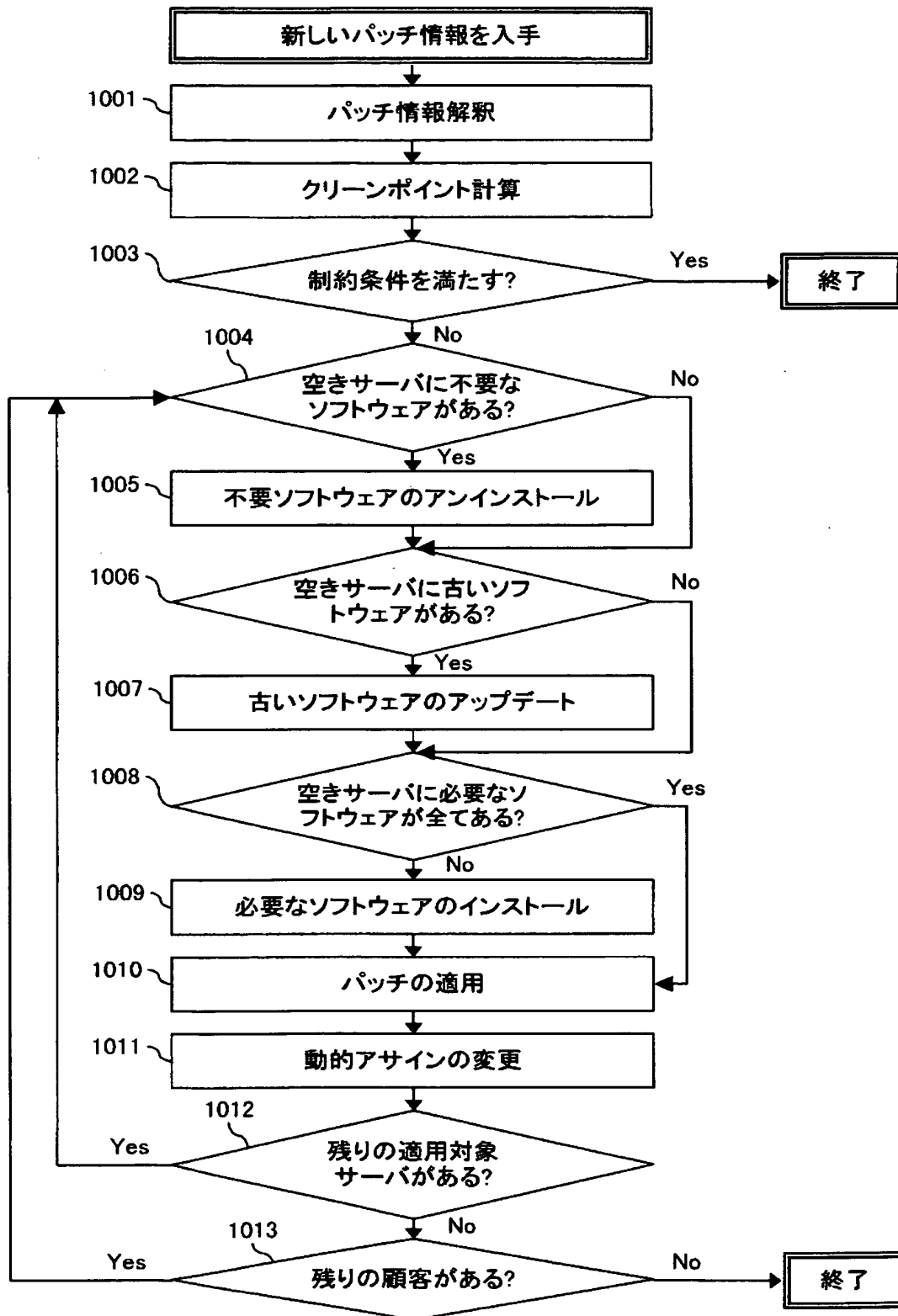
【図 9】

アップデート適用表を示す図

	ソフトウェア名	サーバ名	バージョン	CL
901	Software1	S1	1.3	70
902	Software1	S2	1.4	80
903	Software1	S3	1.5	100
904	Software2	S2	5.3	89
	...			

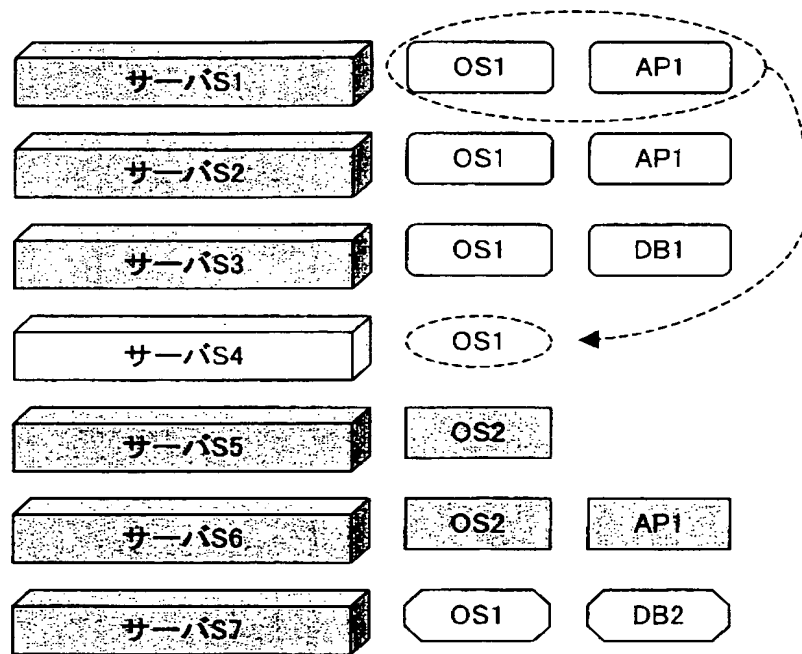
【図 10】

修正適用処理のフローチャート



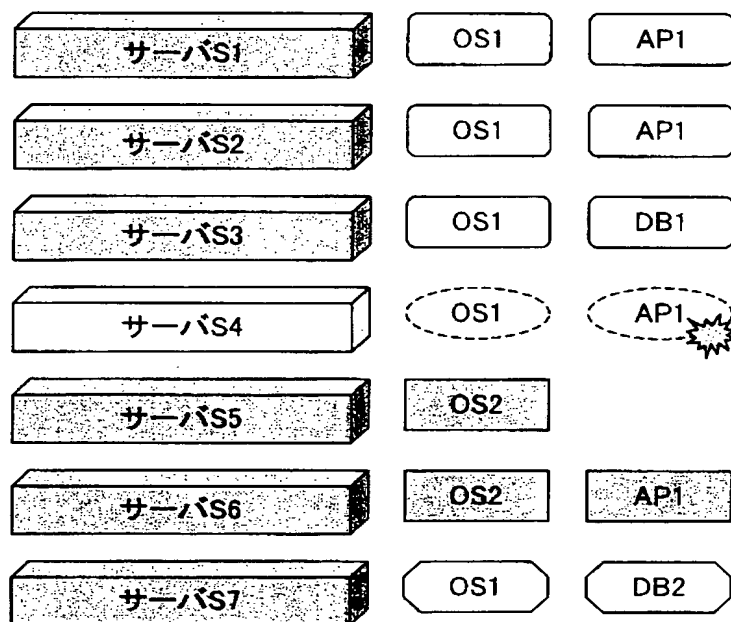
【図 11】

共用サーバプールの第1の状態を示す図



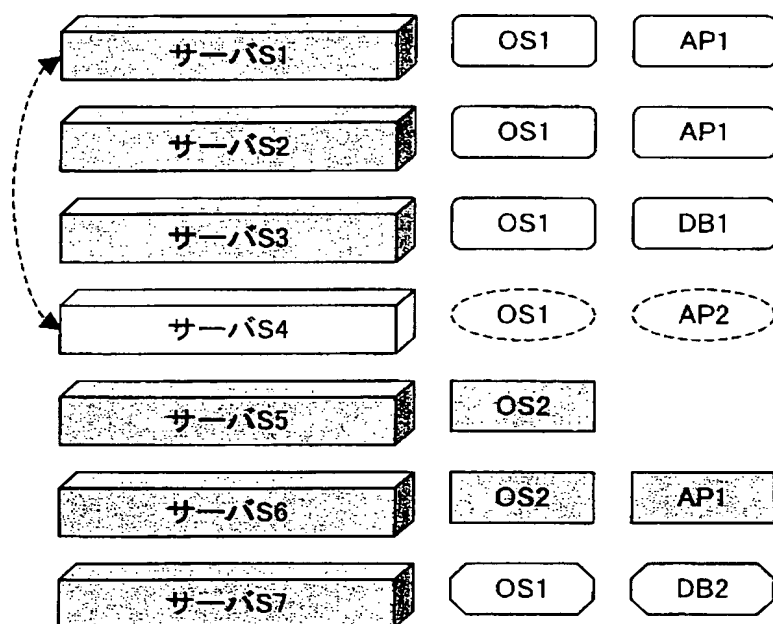
【図 12】

共用サーバプールの第2の状態を示す図



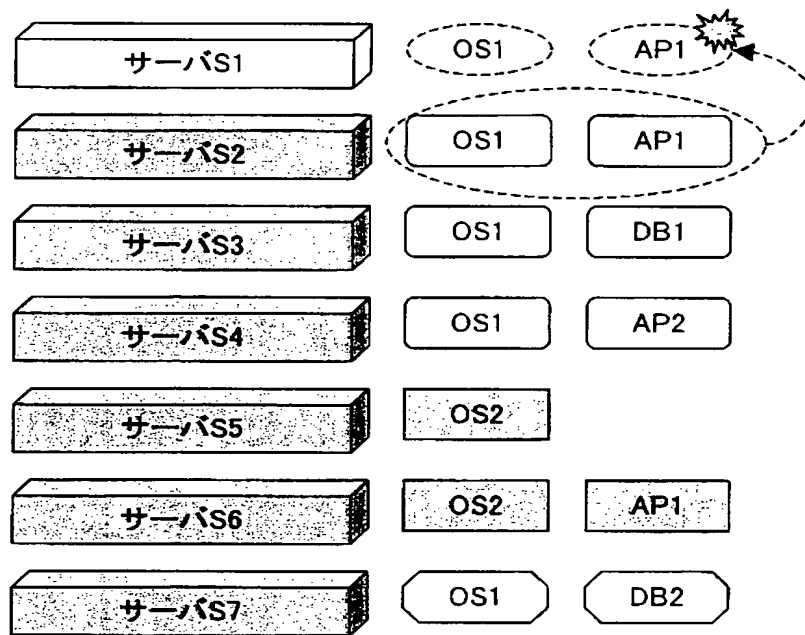
【図 13】

共用サーバプールの第3の状態を示す図



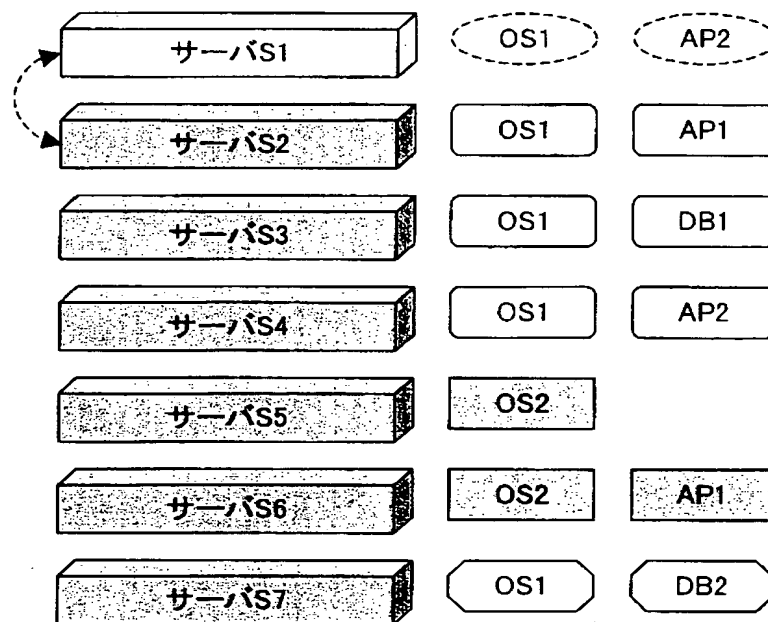
【図 14】

共用サーバプールの第4の状態を示す図



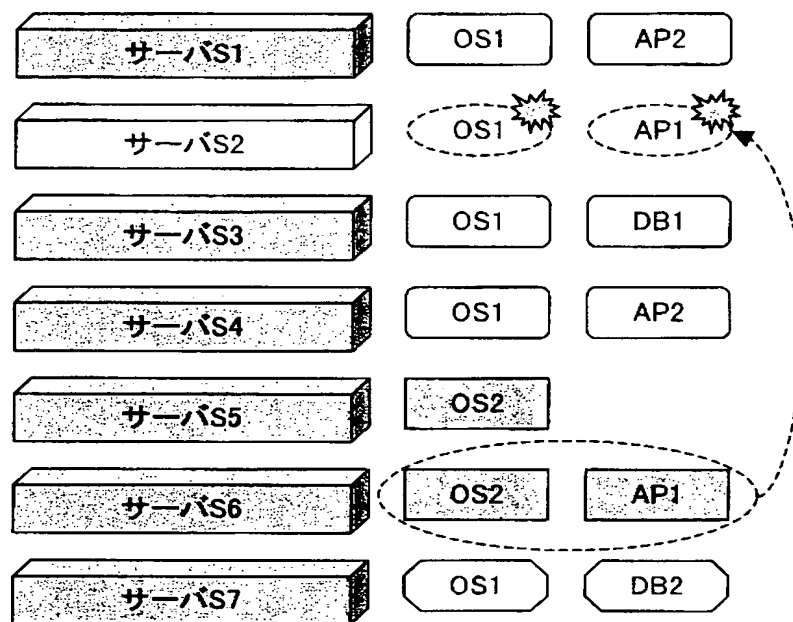
【図 15】

共用サーバプールの第5の状態を示す図



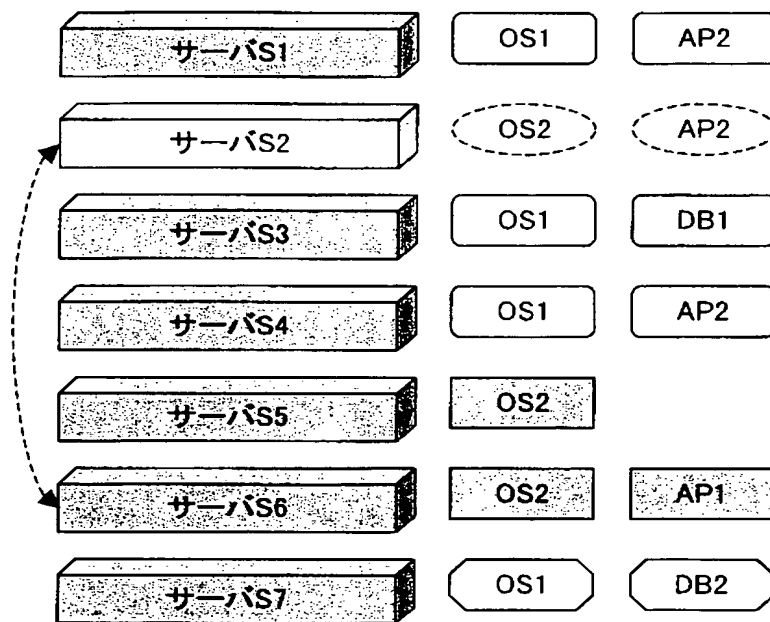
【図 16】

共用サーバプールの第6の状態を示す図



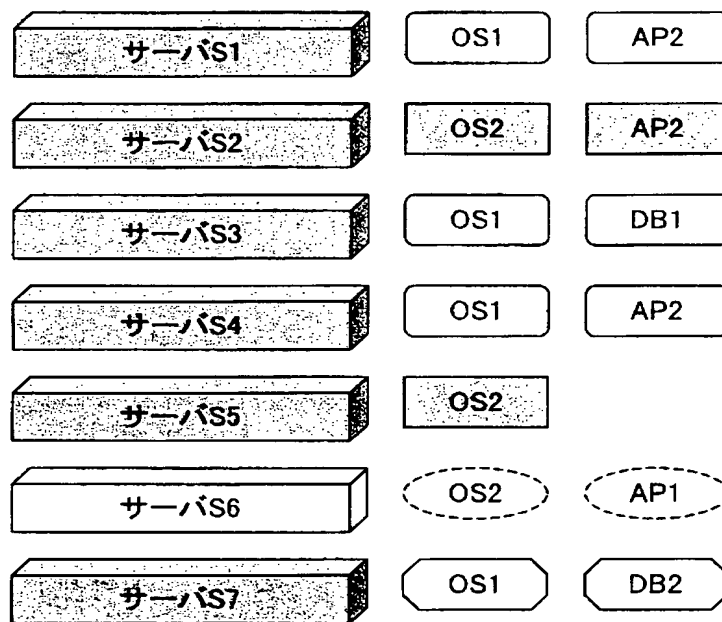
【図 17】

共用サーバプールの第7の状態を示す図



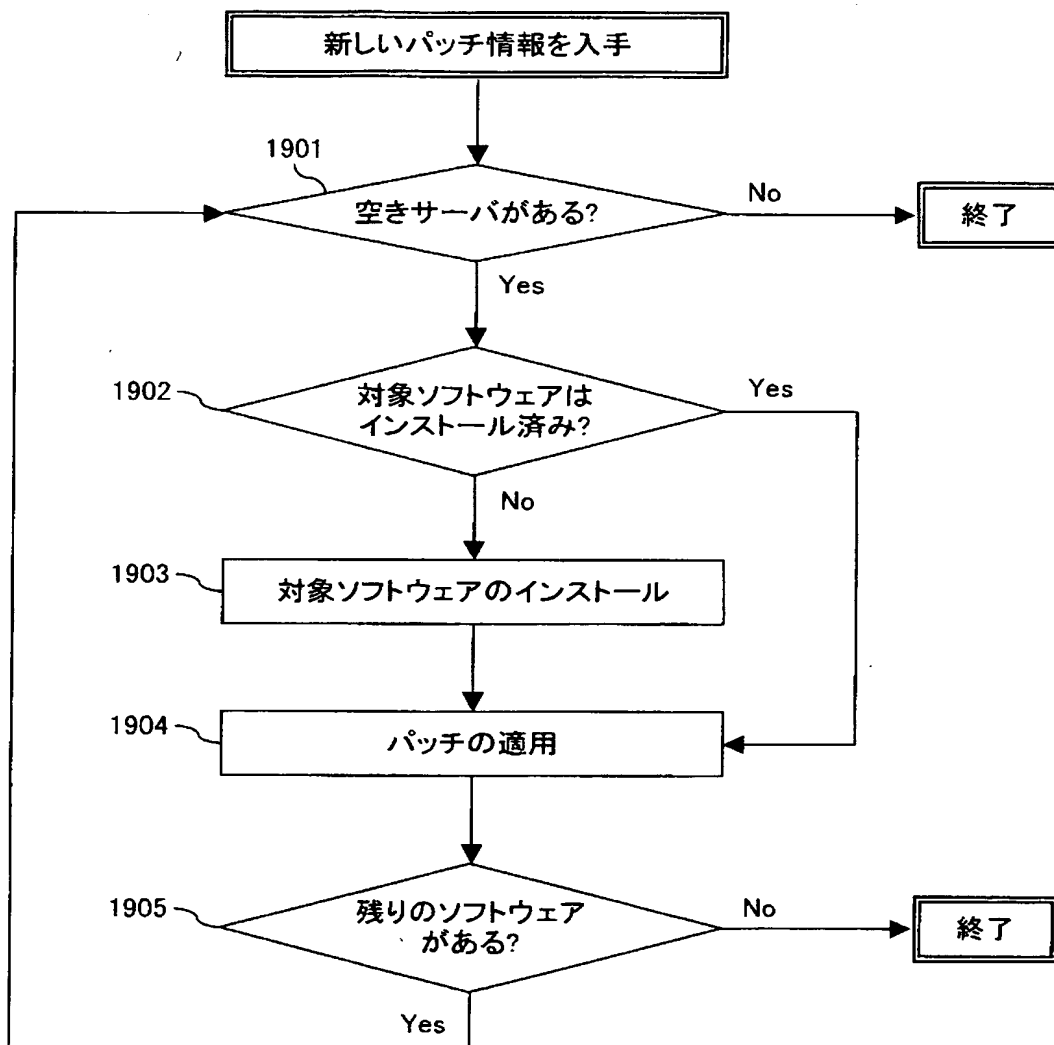
【図 18】

共用サーバプールの第8の状態を示す図



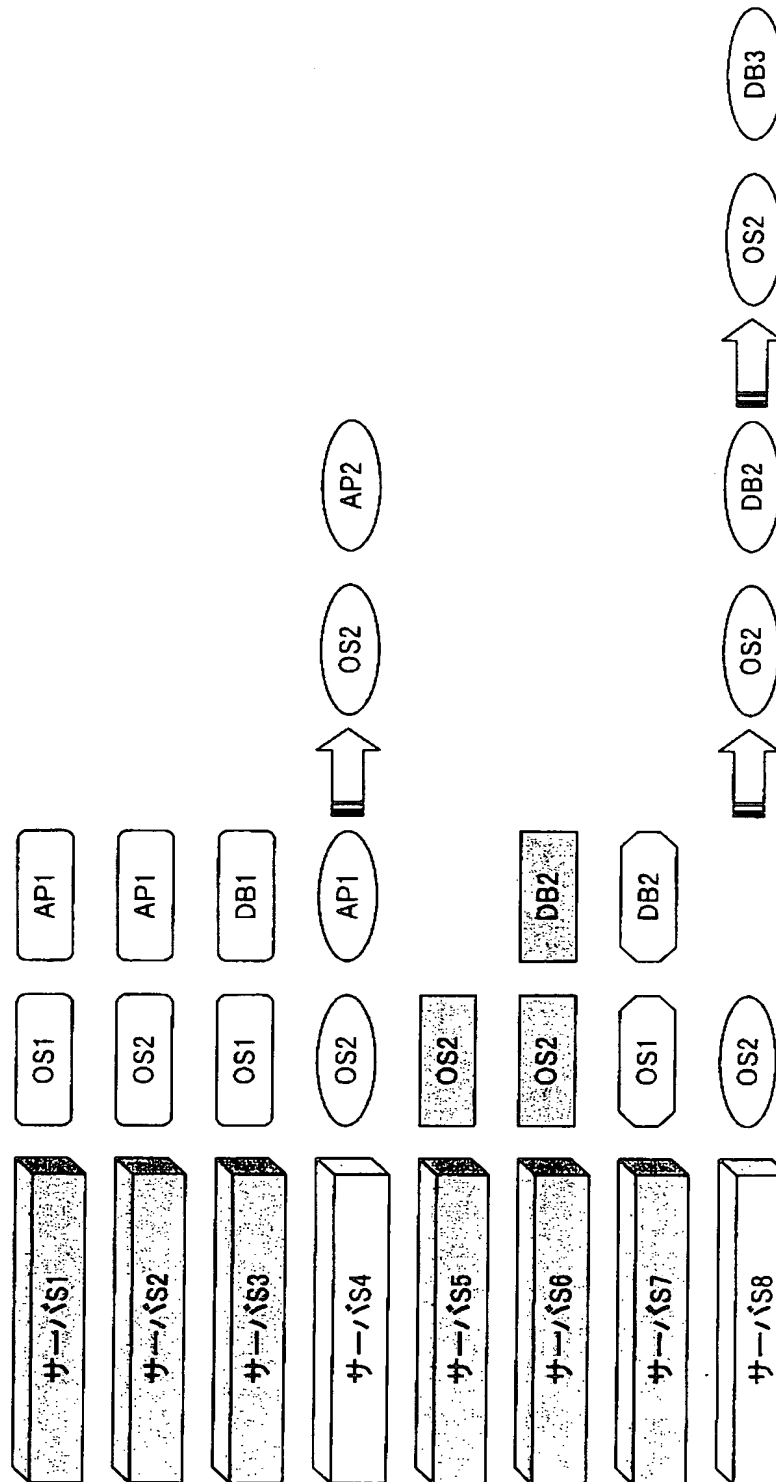
【図 19】

事前修正適用処理のフローチャート



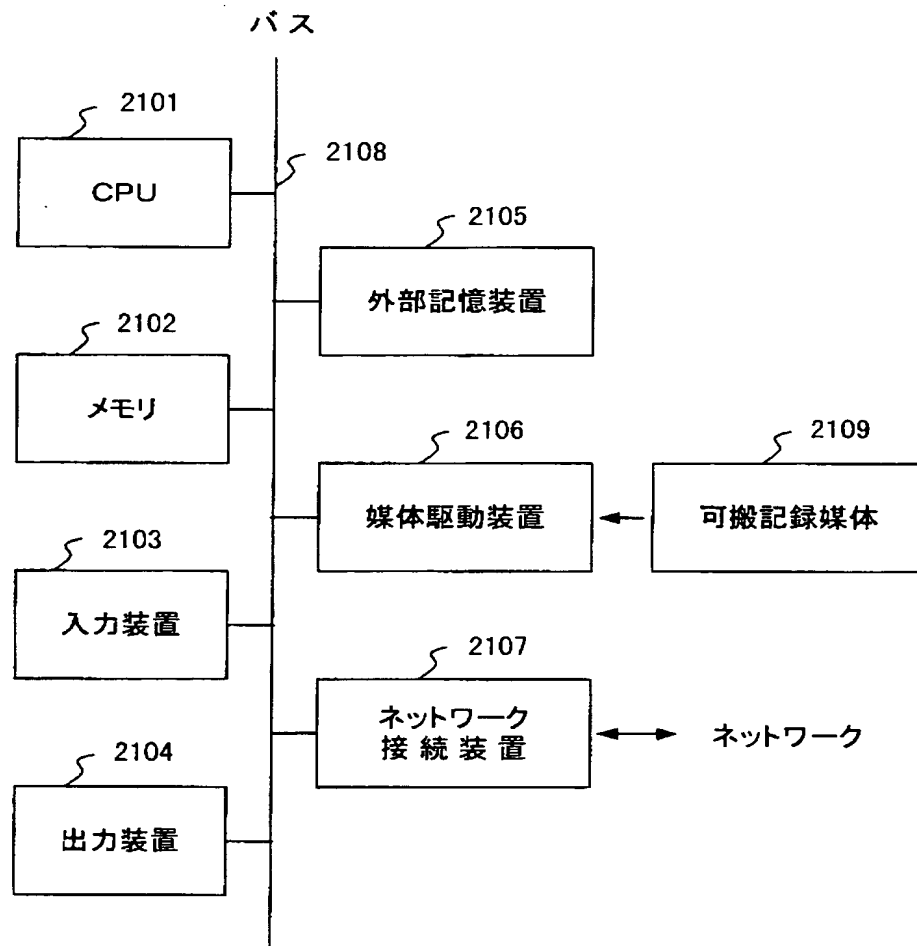
【図 20】

事前修正適用処理を示す図



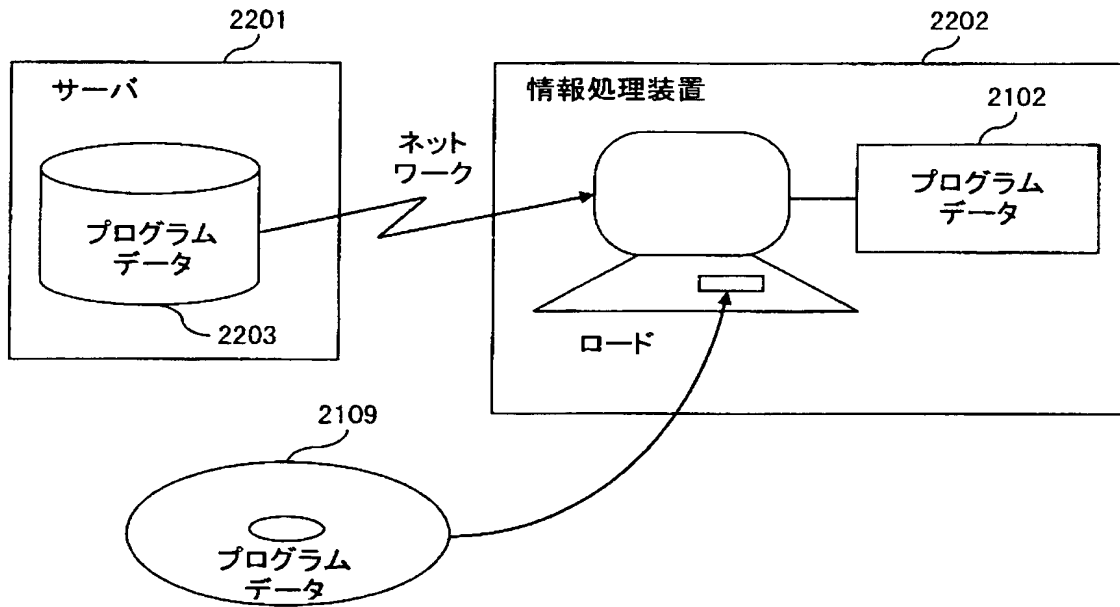
【図 21】

情報処理装置の構成図



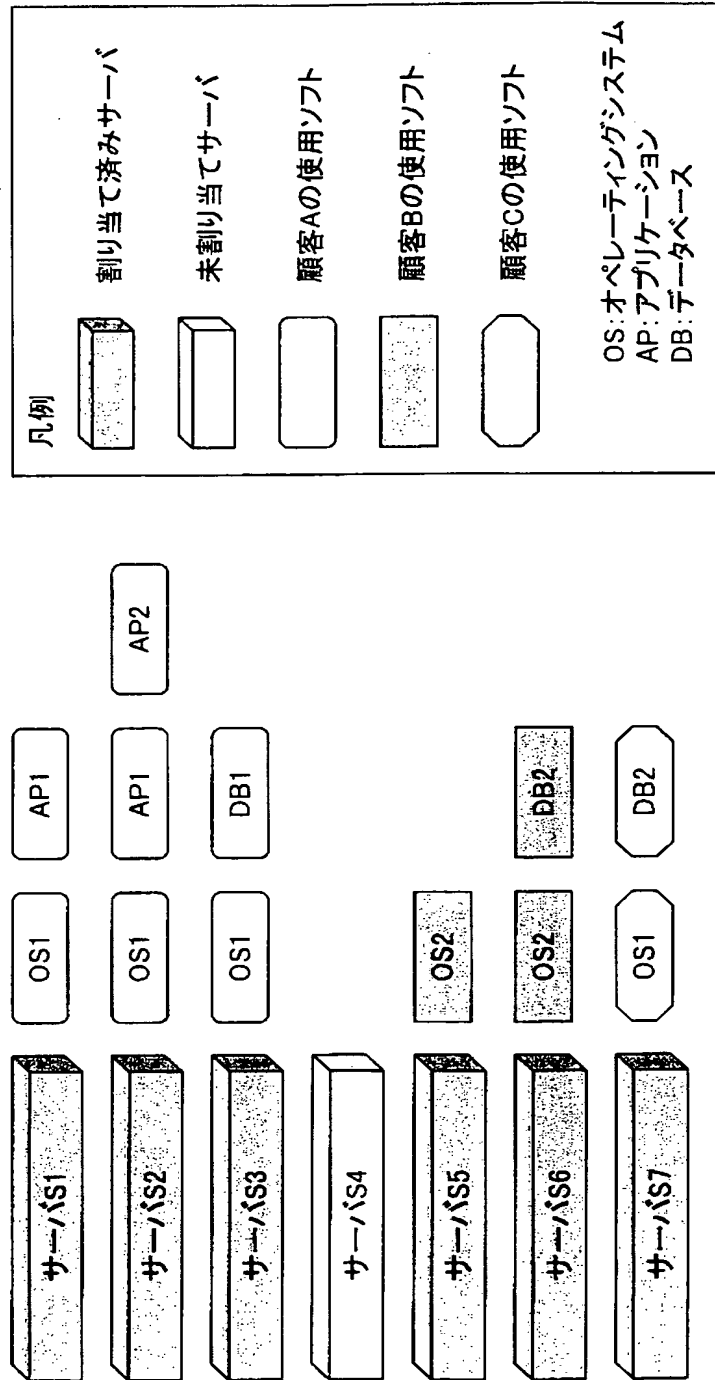
【図 22】

記録媒体を示す図



【図 23】

サーバ共用システムを示す図



【書類名】 要約書

【要約】

【課題】 複数のソフトウェアを複数のサーバに動的に割り当てて顧客サービスを行うシステムにおいて、ソフトウェアに対して修正情報を的確に適用する。

【解決手段】 修正情報の発行時に提供される情報を用いて、顧客が使用しているソフトウェアの新しさとセキュリティレベルを統合した指標を計算し、得られた指標が顧客により指定された制約条件を満たさなければ、そのソフトウェアに修正情報を適用してソフトウェアをアップデートする。

【選択図】 図 1

特願 2 0 0 3 - 3 2 9 0 0 7

出 願 人 履 歴 情 報

識別番号

[0 0 0 0 0 5 2 2 3]

1 . 変更年月日

1 9 9 6 年 3 月 2 6 日

[変更理由]

住所変更

住 所

神奈川県川崎市中原区上小田中 4 丁目 1 番 1 号

氏 名

富士通株式会社